

Forschungsdaten als organische Systeme?

Christoph Rzymski, Robert Forkel, Johann-Mattis List

Max-Planck-Institut für Menschheitsgeschichte
Abteilung Sprach- und Kulturevolution (DLCE)

Ausgangsthese

Der konventionelle Lebenszyklus von
Forschungsdaten sieht Daten an ihrem Endpunkt vor
allem als Ausgangspunkt für die Generierung und
Erhebung neuer Daten.



(<https://www.ianus-fdz.de/lebenszyklus>)

In unserer täglichen Arbeit mit stark vernetzten, reichhaltig annotierten, lexikalischen Daten zeigt sich, dass diese monolithische bzw. starre Sicht auf Daten, die einen Anfang und ein Ende haben (sollen), nicht unserem wissenschaftlichen Alltag entspricht.

Col. 4-6

Tableau I ^a Col. 4-6, N° 1-31	1	2	3	4	5	6
	Il fait	chaud	aujourd'hui	C'est	un bon	temps
	facit	caldu	hödie	est	bōnu	tēmpus
I. Vaud						
1. Chevroux . . .	yē fā*	tsō	wē*	l ē	b bōn*	*tē(i)
2. Vaugondry . . .	fā	tsō	wēl	s ē	bā* bōn	tā
3. L'Auberson . . .	t fō*	tsō	wē*	t fō*	bō	tē*
4. Vallorbe . . .	t fā	*bēi* tsō	wēl*	s ē	b* bō*	tā*
5. Le Sentier. . .	fā	tsō	wēl*	s ēt	bōd* bēōn*	tā*
6. Longirod . . .	ēi fā*	tsō*	wē*	ēi fā*	bōn*	tā*
7. Commugny . . .	ēi fā*	šō	wā	y ē	b bō	tā
8. Vullierens . . .	yē fā*	tsō	*wēd*	yē fā*	b bōn*	tā
9. Arnex . . .	fā	tsō	wēl	—	bōn	tē*
10. Villars-le-Terroir	yē fā	*bēi tsō*	wē*	l ē	b bōn	tē*
11. Prahlins . . .	yē fā	tsō	wē	l ē	b bō	*tē(i)*
12. Montpreveyres .	yē fā	tsō	wā	l ē	bā bōn	*tēi*
13. Charnex . . .	t fā	tsō	*wē*	l ē	b bō	*tē
14. Roche . . .	t fā	tsō	wāe*	l ē	b bō	tē
15. Ormont-Dessus .	yē fā	tsō	wē	y ē	b bō	tē
16. Château-d'Oex .	t fā	tsō*	wē	d ē	b bō	*tē*
II. Valais						
17. Saint-Gingolph .	fō	tsō*	wēl*	y ēt	b bō	tē

(Tableaux phonétiques des patois suisses romands)

Value Set Le Sentier/a good [5]

TPPSR form:

éðv béðv

IPA form:

ເວັບໄຟ

Segments:

е є + б є

Prosodic

w_cv

Dialect:

Le Sentier

a good [E] / up here

Sentences

s et ə̄ññ hə̄ññ tæ

C'est un bon temps

Horse (68).

Number in
General
List.

AGGLUTINATIVE NON-INDIAN LANGUAGES.

Japanese . . .	<i>uma</i>
Ainu . . .	<i>umma</i>
Korean . . .	<i>mâl</i>
Turki . . .	<i>ât</i>
Manchu . . .	<i>morin</i>
Mongolian . . .	<i>morin</i> ; (stallion) <i>âjirγà</i>
Saukpâ . . .	<i>mâ-ri</i>
Basque . . .	<i>zaldi</i> , <i>ala-</i>

UNCLASSED LANGUAGE.

850. Burušaskî . . .	<i>hayur</i>
----------------------	--------------

AUSTRO-NESIAN LANGUAGES.

2. Malay . . .	<i>kudâ</i>
Cham . . .	<i>âsaih</i>
1. Salôñ . . .	<i>mâ*</i>

Number in
General
List.

KAREN LANGUAGES.

35. Pwo, literary . . .	$=\theta\bar{e}$, $-ka =\theta\bar{e}$
" Bassein . . .	θi , $k^i\theta i$
" Maulmein . . .	θe
36. Taungθu . . .	$-ka^=\theta\bar{e}$
34. Sgâ, literary . . .	$k^i\theta e$
" spoken . . .	$k^i\theta e$
32. Bwè . . .	θri
41a. Wewaw . . .	$k^i\theta e$
33. Karenbyu . . .	θiri
Brâ ^o . . .	$\theta\ddot{a} ri$
40. Karenni . . .	$dâ \theta i$
Yintalä . . .	$tâ sî$
Sin-hmâ	
Mäpauk . . .	θai
39. Gheko . . .	θai
37. Padaung . . .	θai

Number in
General
List.

123. Abor
124. Miri
125. Daflâ . . .	<i>g'urâ</i> (Aryan)
126. Mišmi, Digärü . . .	<i>grue</i> (? Aryan)
Mijû . . .	<i>kom-beñ</i>

Lolo-Mos'o Group.

Si-hia . . .	<i>lin lo, riñ ro</i>
273. Lolo, /Ñi . . .	$-m^u$
A-hi . . .	<i>mo^o, -a -lō m^o</i>
Lo-lo p'o . . .	<i>mu^o</i>
276. A-ka (Kâ) . . .	<i>md̪</i>
277a. A-kö . . .	<i>mē pâ</i>
275. Lisu . . .	$-\ddot{a} -m\ddot{u}$
Lisâ or Yâyin . . .	<i>āmu</i>
274. Mo-s'o . . .	<i>ža</i>
Lahu . . .	<i>tiri, imu</i>
277a. Pyen or Pyin . . .	<i>āmdñ bū</i>

(Linguistic Survey of India)



STEDT

Sino-Tibetan Etymological Dictionary and Thesaurus

(STEDT)

[Изменить параметры просмотра](#)
[Перейти к английской версии](#)

Базы данных

Show only those databases: (All) [nostratic](#) [afro-asiatic](#) [caucasian](#) [austroasiatic](#)

		просмотр		поиск	описание		tree	2006-05-28
<input type="checkbox"/>	Глобальные этимологии Составитель: Старостин С. А.							
<input type="checkbox"/>	Ностратическая этимология Составитель: Старостин С. А.							
<input checked="" type="checkbox"/>	Индоевропейская этимология Составитель: Николаев С. Л.							
<input checked="" type="checkbox"/>	Алтайская этимология Составитель: Старостин С. А.							
	Уральская этимология Составитель: Старостин С. А.							
	Картвельская этимология Составитель: Старостин С. А.							
<input checked="" type="checkbox"/>	Дравидийская этимология Составитель: Старостин Г. С.							
<input checked="" type="checkbox"/>	Чукотско-камчатская этимология Составитель: Мудрак О. А.							
	Эскимосская этимология							

(StarLing/Tower of Babel)

Und vor allem auch:

- (Feldforschungs-)Daten Forschender am Institut
- Daten und Materialien von weltweit verteilten Beitragenden und Mitwirkenden

Durch eine Reihe von Umständen wird fast jeder Abschnitt des Datenlebenszyklus zu einem *moving target*:

- **Erstellung:** permanenter input neuer Daten, *lifting* historischer Daten, Anreicherung alter Daten, variable Klassifizierungen
- **Verarbeitung:** Anforderungsprofile sind permanent im Wandel, siehe **Erstellung**
- **Analyse:** neue Anforderungen an Daten führen zu neuen und verbesserten Analyseschritten, siehe **Verarbeitung**

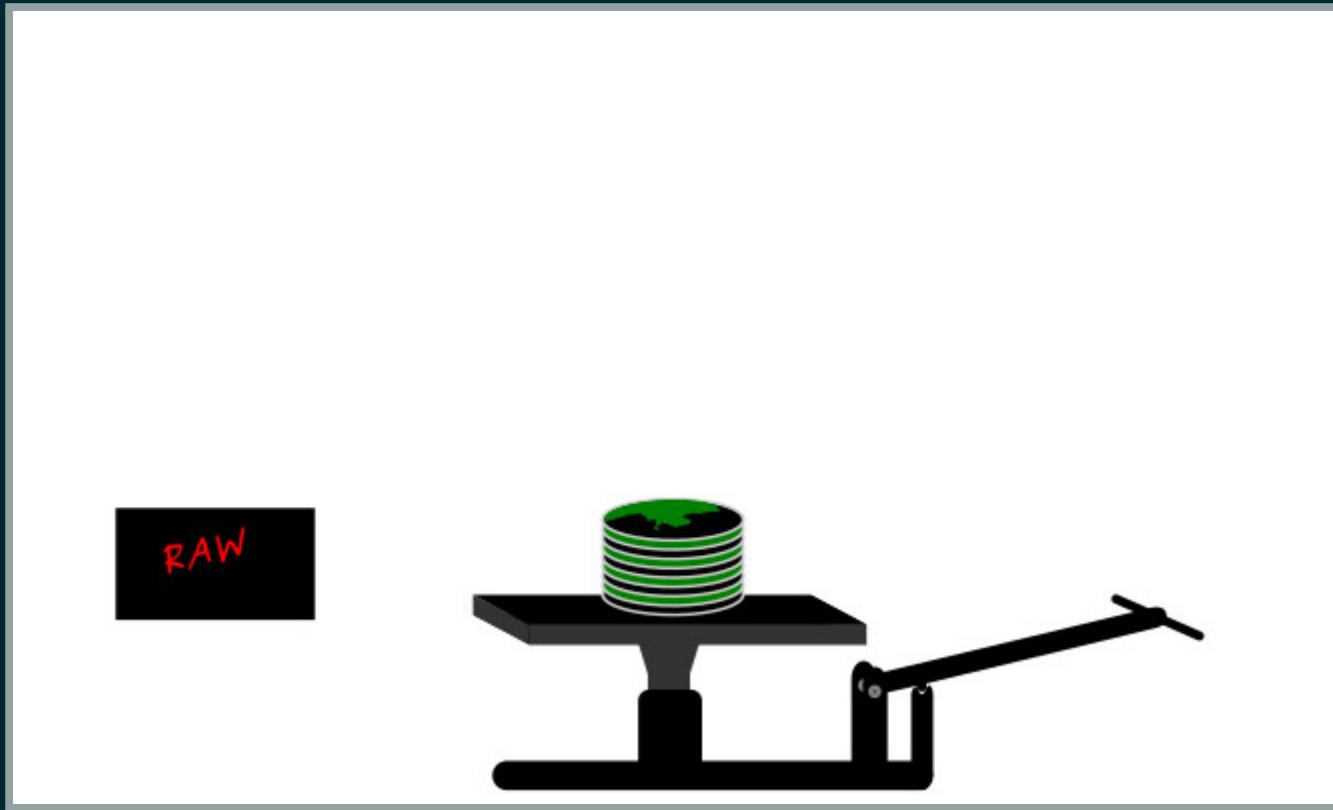
Durch eine Reihe von Umständen wird fast jeder Abschnitt des Datenlebenszyklus zu einem *moving target*:

- **Archivierung:** neue Zwischenergebnisse müssen vorgehalten werden, siehe **Analyse**
- **Zugang:** Zugriff auf Zwischenschritte muss flexibel und konfigurierbar sein, siehe **Archivierung**
- **Nachnutzung:** welche Konfiguration welcher Versionen verwendet worden ist, muss transparent und nachnutzbar sein, siehe **Zugang**

Zusammenfassung

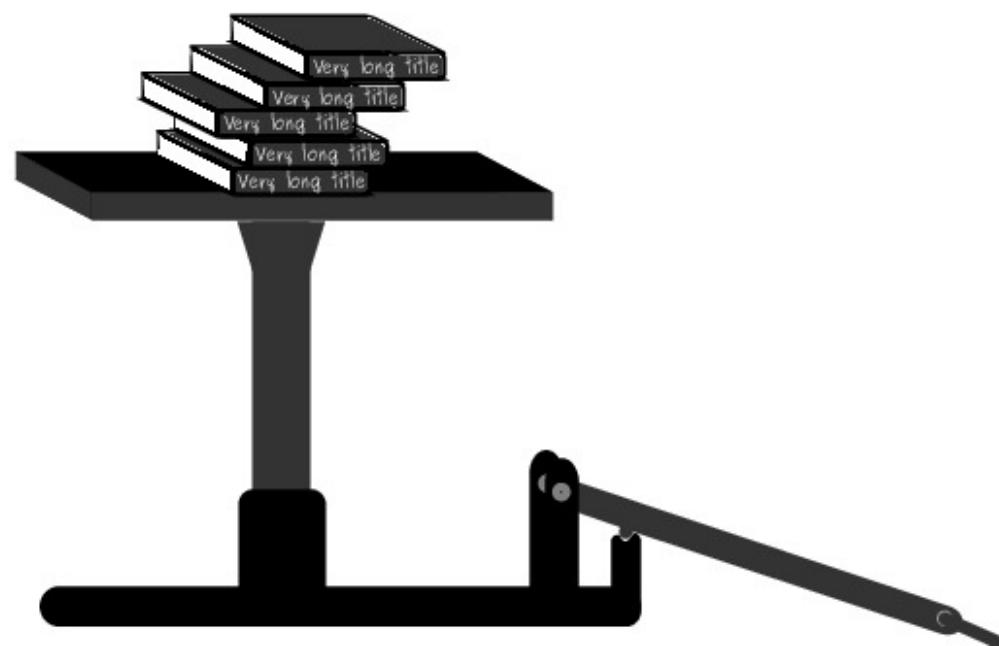
- Daten befinden sich während ihrer gesamten Lebensphase in einem Kontinuum
- klar definierte Anfangs- und Endpunkte sind durch sich stets ändernde Datenlagen schwer bis gar nicht zu definieren

Fazit: Verschiedene Bausteine und Komponenten sind nötig, um diesen Umständen im Sinne der guten wissenschaftlichen Praxis zu begegnen.



analogous

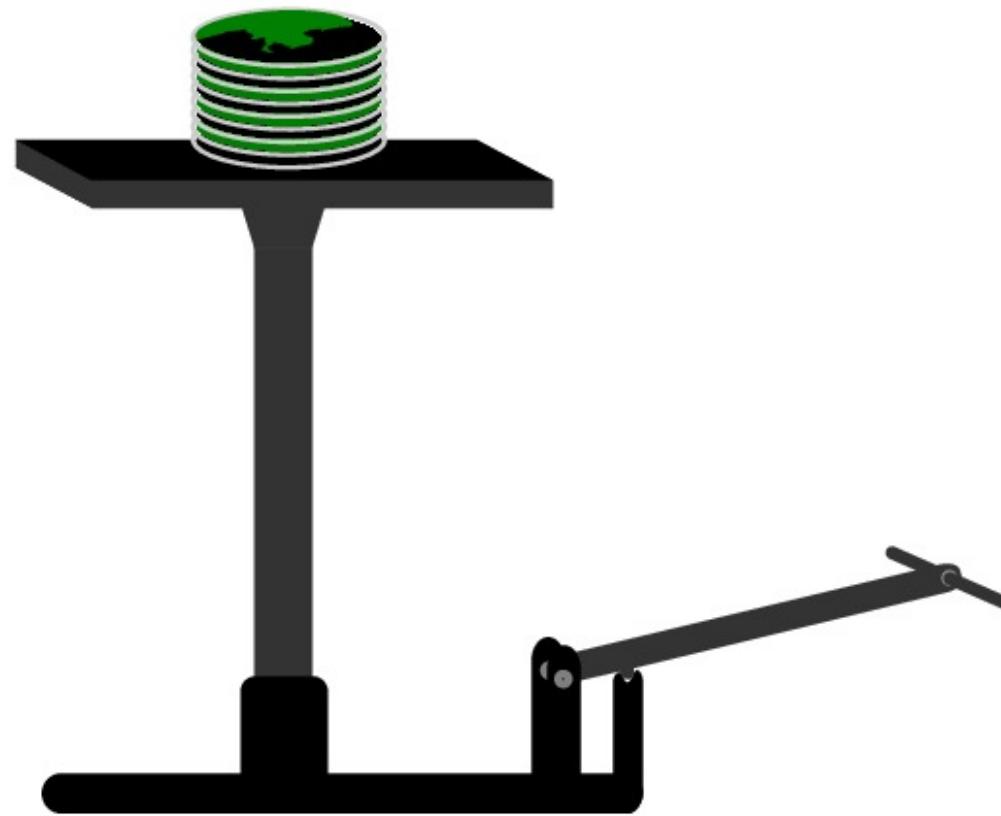
RAW



digital

analogous

RAW





digital

analogous

RAW

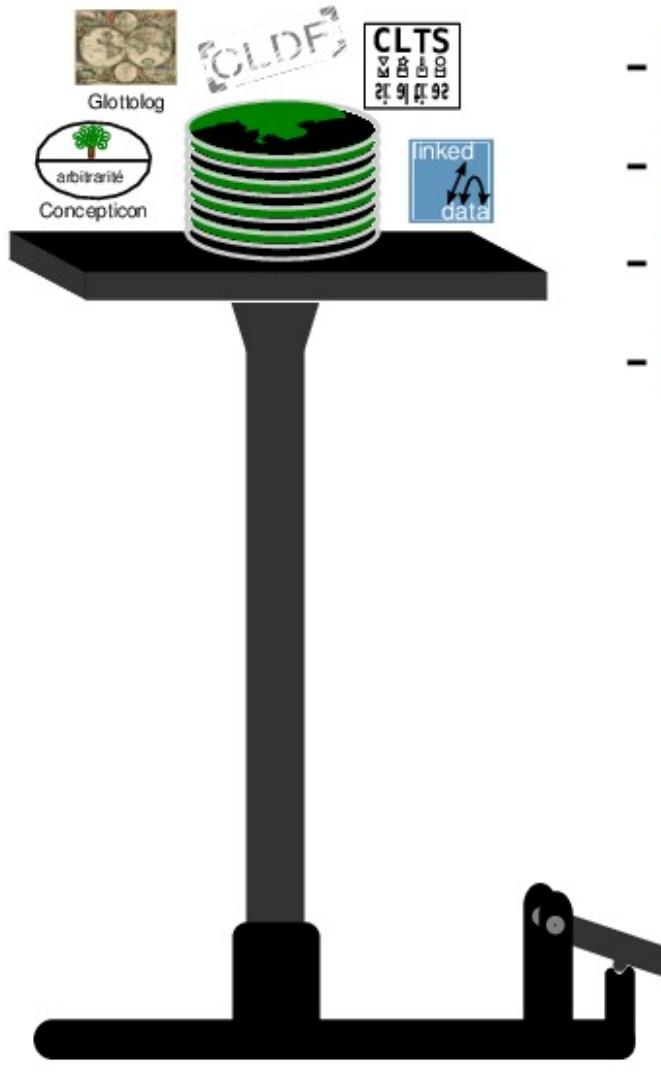




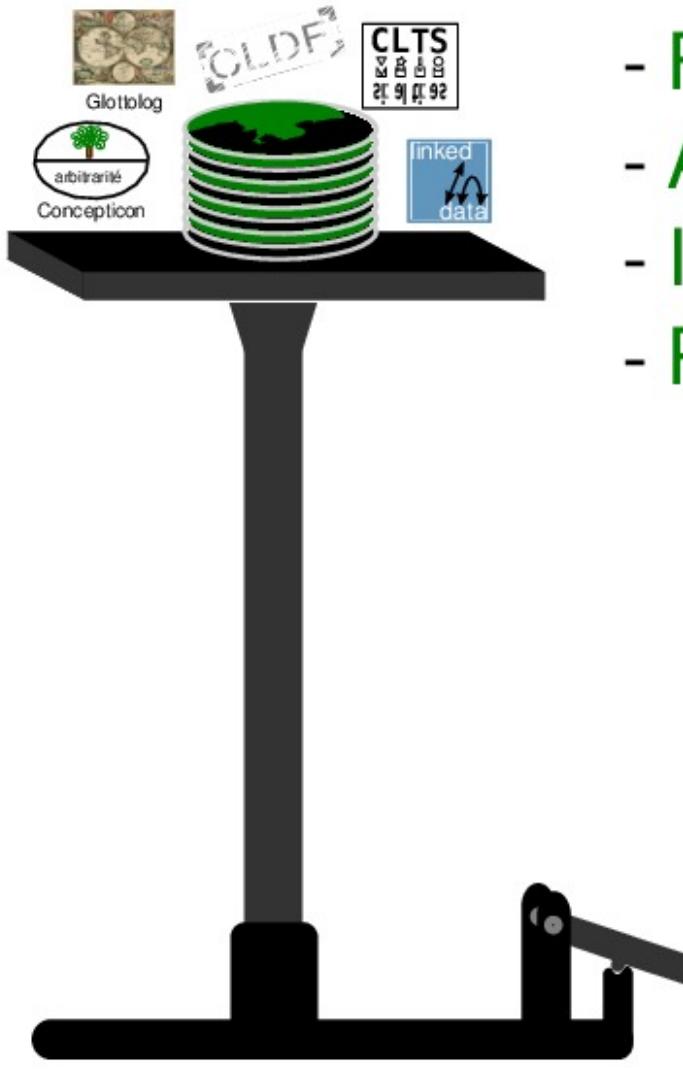
digital

analogous

RAW



- Findable
- Accessible
- Interoperable
- Reusable



- Findable
- Accessible
- Interoperable
- Reusable



Bausteine

1. CLDF, menschen- und maschinenlesbare Formate
2. Versionskontrolle für Daten
3. Referenzkataloge, verteilte Daten, Metadaten
4. Testsuiten für Daten
5. Nutzbarkeit von Daten während ihres gesamten Lebenszykluses

(1) CLDF, menschen- und
maschinenlesbare Formate

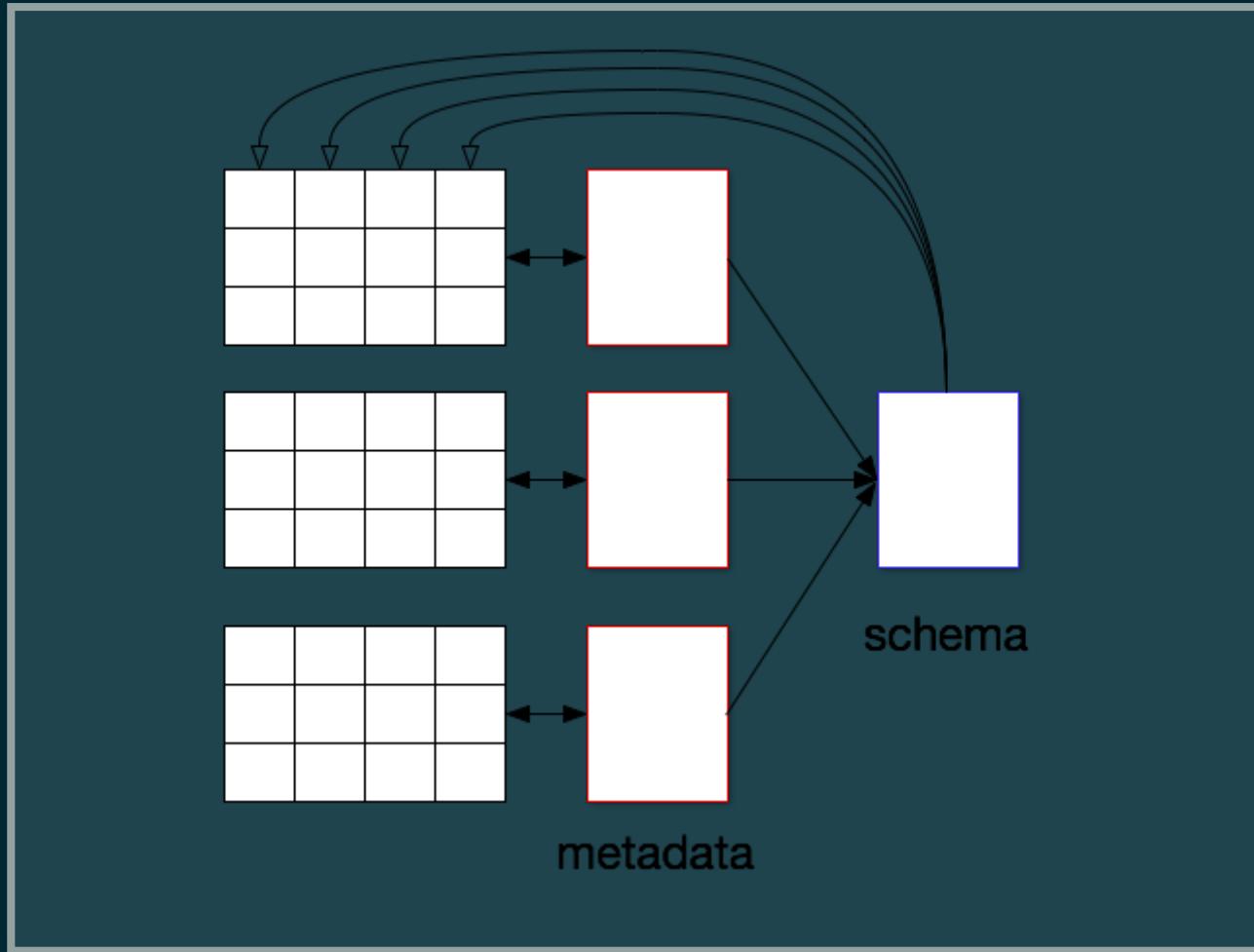
CLDF (Cross-Linguistic Data Formats,
<https://doi.org/10.1038/sdata.2018.205>) ist ein Paketformat zur Beschreibung linguistischer bzw., allgemeiner, tabellarischer Daten.

CLDF orientiert sich stark an den W3C Spezifikationen zu CSV on the Web.

Ein CLDF Datensatz besteht im Kern aus:

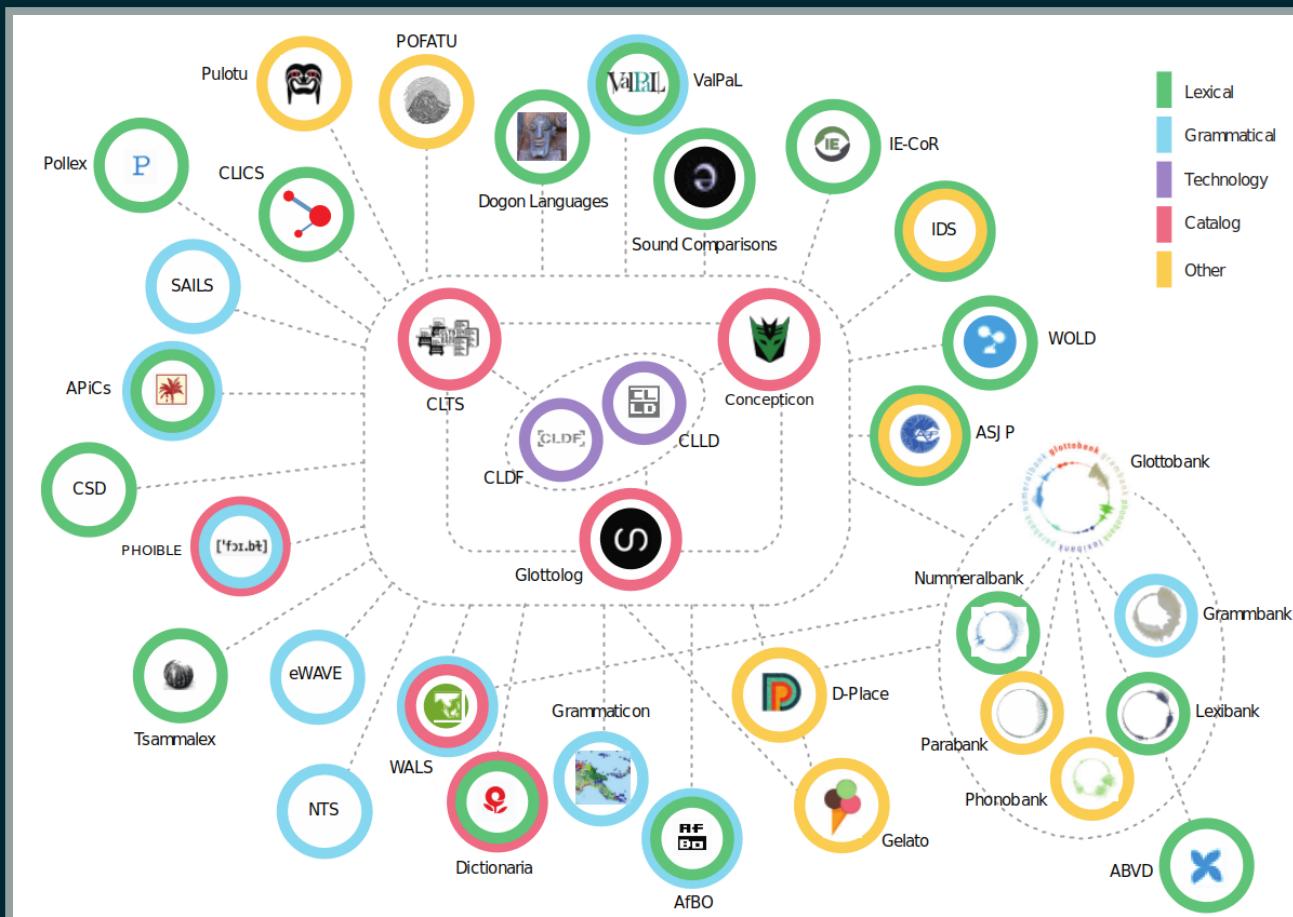
- einer Menge von UTF-8 kodierten Dateien
- einer **CSVW TableGroup** (Metadaten)
- einer `dc:conformsTo`-Eigenschaft für die einzelnen Komponenten welche ihre Semantik aus der CLDF-Ontologie beziehen

```
chrzyki@razorback:~/Repositories/lexibank/halenepal/cldf$ ll
total 2996
drwxrwxr-x 2 chrzyki chrzyki    4096 Apr 22 15:55 .
drwxrwxr-x 8 chrzyki chrzyki    4096 Apr 22 15:51 ..
-rw-rw-r-- 1 chrzyki chrzyki   10283 Apr 22 15:55 cldf-metadata.json
-rw-rw-r-- 1 chrzyki chrzyki 2778069 Apr 22 15:55 forms.csv
-rw-rw-r-- 1 chrzyki chrzyki      19 Apr  8 09:11 .gitattributes
-rw-rw-r-- 1 chrzyki chrzyki    1198 Apr 22 15:55 languages.csv
-rw-rw-r-- 1 chrzyki chrzyki   37899 Apr 22 15:55 parameters.csv
-rw-rw-r-- 1 chrzyki chrzyki     82 Apr  8 09:11 README.md
-rw-rw-r-- 1 chrzyki chrzyki     796 Apr 22 15:55 requirements.txt
-rw-rw-r-- 1 chrzyki chrzyki     256 Apr 22 15:55 sources.bib
-rw-rw-r-- 1 chrzyki chrzyki 204597 Apr 22 15:55 .transcription-report.json
```



(<https://theodi.github.io/presentations/2014-07-15-csv-wg.html>)

CLDF steht damit im Zentrum:



CLDF bedient sich dabei der üblichen Bausteine der W3C zu Daten die im *semantic web* existieren:

- CSV(W): CSV on the web
- JSON-LD: JavaScript Object Notation for Linked Data
- RDF: Resource Description Framework
- Vokabularen wie dem Dublin Core Schema
- eine Ontologie

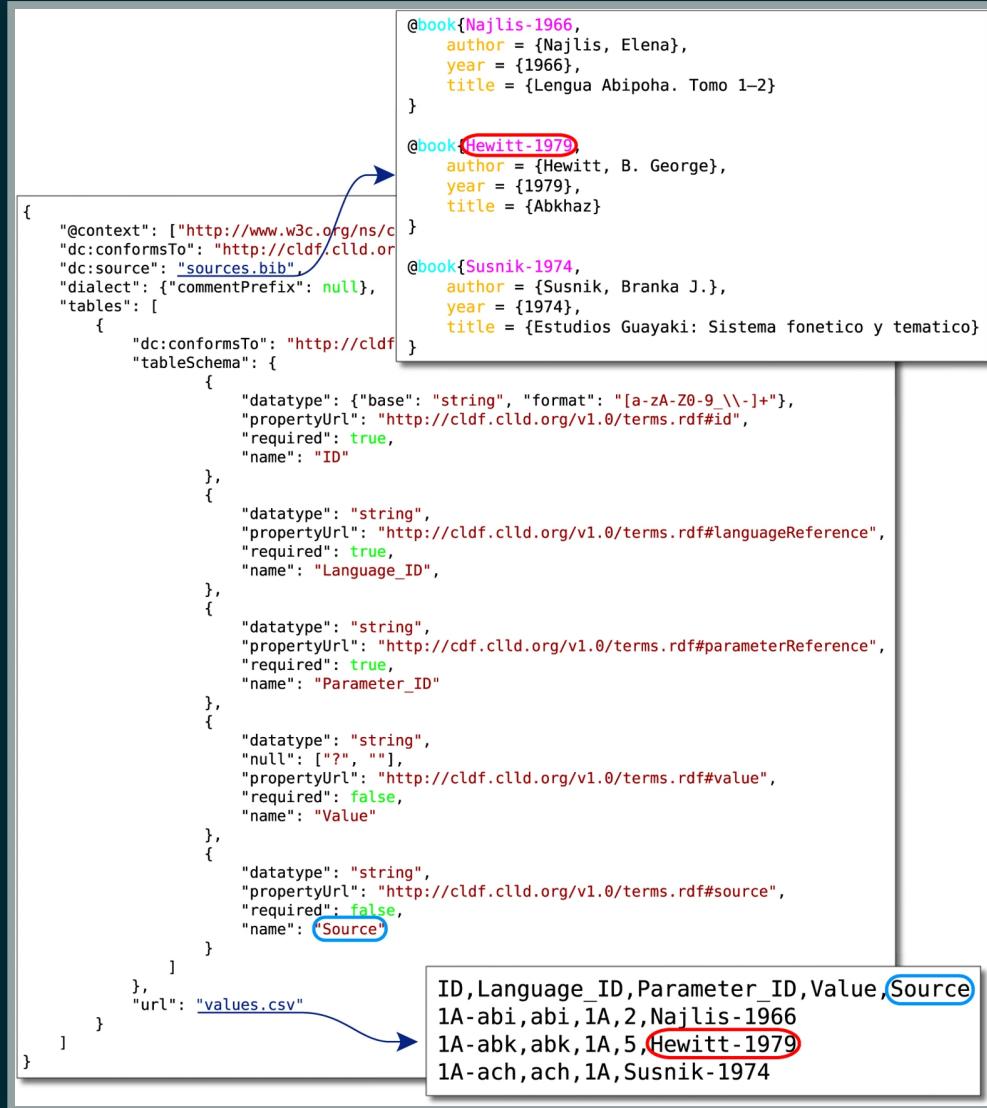
FAIRness ist damit in CLDF nicht nur ein Ziel sondern fundamentaler Bestandteil des Designs sowie der einzelnen Komponenten.

CLDF bietet somit:

- ein CSV-basiertes Paketformat mit Semantik über die CLDF-Ontologie, verschiedenen Modulen für verschiedene Anwendungsszenarien, eingebauter Unterstützung für Referenzkataloge
- wiederverwendbare und stabile *Identifiers* für Sprachinformationen, semantische Konzepte, Transkriptionssystemen und Strukturfeatures

CLDF bietet somit:

- textbasierte und relationale Daten die nicht auf externe Software angewiesen sind (davon aber profitieren können)
- eine Vielzahl von Interfaces, Werkzeugen, und APIs



(2) Versionskontrolle für Daten

Um Daten als moving target greifbar zu machen,
sollten:

- Zwischenresultate eindeutig identifizierbar sein:
 - Versionsnummern, Releases, Zenodo
- Daten einfach zugänglich sein:
 - Daten als installierbare Pythonpakete
- Daten mittels üblicher Arbeitsschritte von
Versionskontrolle kuratiert werden
 - pull requests, commits, issues

Releases:

Latest release

v0.11
dd87a6a

Compare ▾

TuLeD: Tupían lexical database

xrotwang released this on 23 Mar

Cite as

Fabrício Ferraz Gerardi, Stanislav Reichert, Carolina Aragon, Johann-Mattis List, & Tim Wientzek. (2021). TuLeD: Tupían lexical database (Version v0.11) [Data set]. Zenodo. <http://doi.org/10.5281/zenodo.4629306>

DOI [10.5281/zenodo.4629306](http://doi.org/10.5281/zenodo.4629306)

▼ Assets 2

Source code (zip)

Source code (tar.gz)

Zenodo:

March 23, 2021

Dataset Open Access

TuLeD: Tupán lexical database

Fabrício Ferraz Gerard; Stanislav Reichert; Carolina Aragon; Johann-Mattis List; Tim Wientzek

Data curator(s)

Robert Forkel

Other(s)

Tiago Tresoldi

TuLeD: Tupán lexical database

Preview

tuled-v0.11.zip

- tupian-language-resources-tuled-dd87a6a
 - .github
 - workflows
 - python-package.yml
 - gitignore
 - zenodo.json
 - CONTRIBUTORS.md
 - FORMS.md
 - LICENSE
 - README.md
 - RELEASING.md
 - TRANSCRIPTION.md
 - cldf
 - .gitattributes
 - transcription-report.json
 - cldf-metadata.json
 - cognates.csv
 - forms.csv

Available in

88 views 3 downloads

See more details...

Indexed in

GitHub

OpenAIRE

Publication date: March 23, 2021

DOI: DOI 10.5281/zenodo.4629306

Keyword(s): cldf:Wordlist, linguistics

Related identifiers: Supplement to <https://github.com/tupian-language-resources/tuled/tree/v0.11>

Communities: Lexibank, TULaR: Tupán Language Resources

License (for files): Creative Commons Attribution 4.0 International

Files (9.0 MB)

Name	Size
tupian-language-resources/tuled-v0.11.zip	9.0 MB
md5:85493ecf0e1ff9ad7a63316cc9ac8548	

Preview Download

Git und GitHub:

Code Issues Pull requests Actions Projects Wiki Security Insights

master 6 branches 2 tags Go to file Add file Code

File / Commit	Description	Date
xrotwang	completed handling of sources.bib via git submodule; re-ran workflow	ab25738 on 26 Mar
.github/workflows	re-run download and makecldf	2 months ago
cldf	completed handling of sources.bib via git submodule; re-ran workflow	last month
etc	added short description of the curation workflow	last month
raw	completed handling of sources.bib via git submodule; re-ran workflow	last month
.gitignore	new tuled version	6 months ago
.gitmodules	added short description of the curation workflow	last month
.zenodo.json	updated release instructions - closes tupian-language-resources/pytul...	last month
CONTRIBUTING.md	added short description of the curation workflow	last month
CONTRIBUTORS.md	updated release instructions - closes tupian-language-resources/pytul...	last month
FORMS.md	update	6 months ago
LICENSE	update	8 months ago
README.md	completed handling of sources.bib via git submodule; re-ran workflow	last month
RELEASING.md	updated release instructions - closes tupian-language-resources/pytul...	last month
TRANSCRIPTION.md	completed handling of sources.bib via git submodule; re-ran workflow	last month
errors.md	completed handling of sources.bib via git submodule; re-ran workflow	last month
languages.geojson	having run the cldfbench	2 months ago
lexibank_tuled.py	completed handling of sources.bib via git submodule; re-ran workflow	last month
mapNimu2.png	Add files via upload	10 months ago
metadata.json	rc1 for release v0.11	last month

About TuLeD: Tupán lexical database
Readme CC-BY-4.0 License

Releases 2 TuLeD: Tupán lexical database (Latest) on 23 Mar + 1 release

Contributors 7

Languages TeX 84.0% Python 16.0%

Daten als (installierbare) Pakete:

30 lines (30 sloc) | 2.35 KB

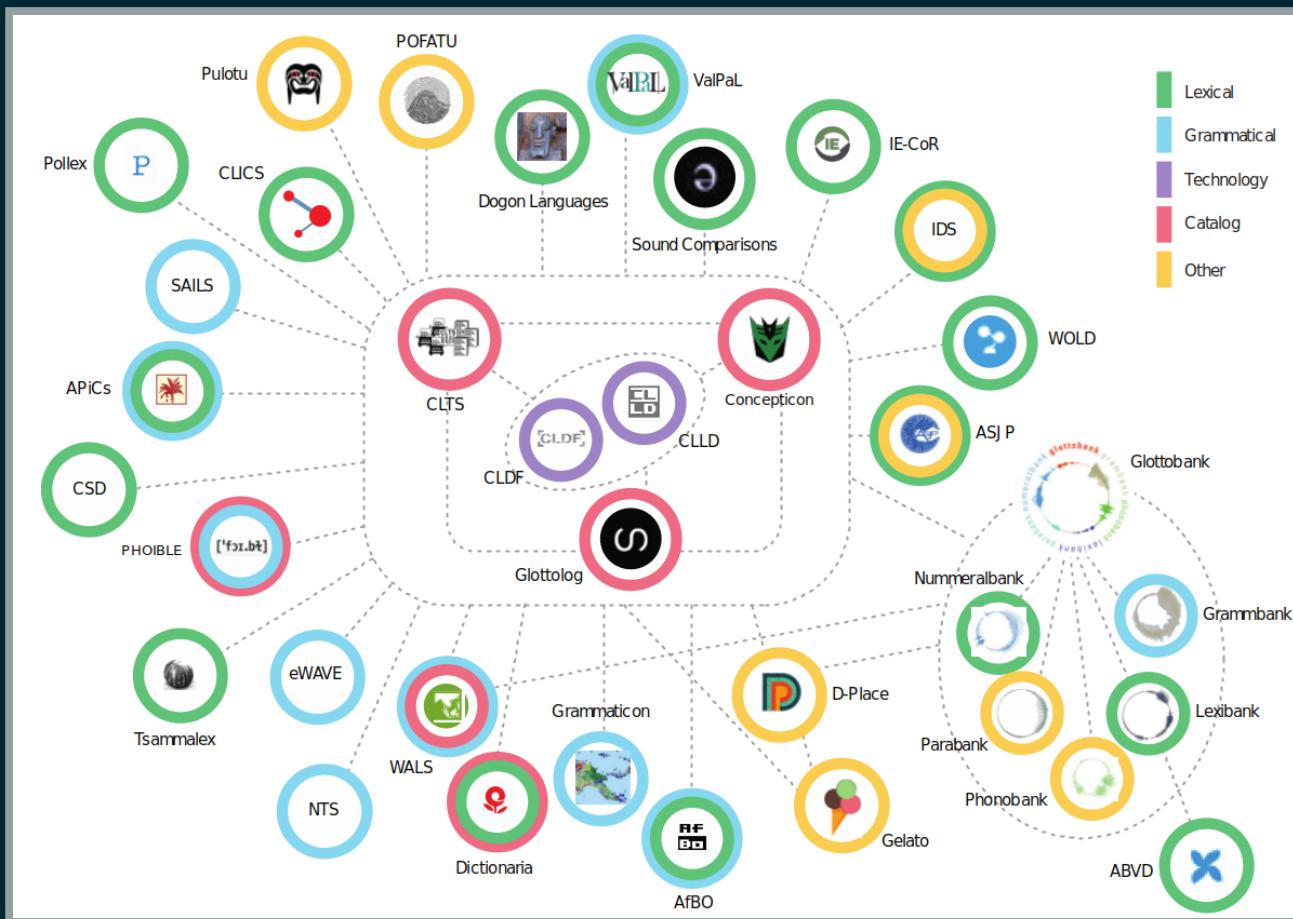
Raw Blame   

```
1 -e git+https://github.com/lexibank/logos.git@v3.0#egg=lexibank_logos
2 -e git+https://github.com/lexibank/allenbai.git@v3.0#egg=lexibank_allenbai
3 -e git+https://github.com/lexibank/bantubvd.git@v3.0#egg=lexibank_bantubvd
4 -e git+https://github.com/lexibank/beidasinitic.git@v4.0#egg=lexibank_beidasinitic
5 -e git+https://github.com/lexibank/bowernpny.git@v3.0#egg=lexibank_bowernpny
6 -e git+https://github.com/lexibank/hubercolumbian.git@v3.0#egg=lexibank_hubercolumbian
7 -e git+https://github.com/lexibank/ids.git@v3.0#egg=lexibank_ids
8 -e git+https://github.com/lexibank/kraftchadic.git@v3.0#egg=lexibank_kraftchadic
9 -e git+https://github.com/lexibank/northeuralex.git@v3.0#egg=lexibank_northeuralex
10 -e git+https://github.com/lexibank/robinsonap.git@v3.0#egg=lexibank_robinsonap
11 -e git+https://github.com/lexibank/satterthwaitetb.git@v3.0#egg=lexibank_satterthwaitetb
12 -e git+https://github.com/lexibank/suntb.git@v3.0#egg=lexibank_suntb
13 -e git+https://github.com/lexibank/tls.git@v3.0#egg=lexibank_tls
14 -e git+https://github.com/lexibank/tryonsolomon.git@v3.0#egg=lexibank_tryonsolomon
15 -e git+https://github.com/lexibank/wold.git@v3.0#egg=lexibank_wold
16 -e git+https://github.com/lexibank/zgraggenmadang.git@v3.0#egg=lexibank_zgraggenmadang
17 -e git+https://github.com/lexibank/abrahammonpa.git@v2.0#egg=lexibank_abrahammonpa
18 -e git+https://github.com/lexibank/bodtkhobwa.git@v2.0#egg=lexibank_bodtkhobwa
19 -e git+https://github.com/lexibank/castrosui.git@v2.0#egg=lexibank_castrosui
20 -e git+https://github.com/lexibank/chenhmongmien.git@v2.0.1#egg=lexibank_chenhmongmien
21 -e git+https://github.com/lexibank/diacl.git@v2.0#egg=lexibank_diacl
22 -e git+https://github.com/lexibank/halenepal.git@v2.0#egg=lexibank_halenepal
23 -e git+https://github.com/lingpy/language-island-paper.git@v3.0.1#egg=lexibank_hantganbangime
24 -e git+https://github.com/lexibank/marrisonnaga.git@v2.0#egg=lexibank_marrisonnaga
25 -e git+https://github.com/lexibank/mitterhoferbena.git@v2.0#egg=lexibank_mitterhoferbena
26 -e git+https://github.com/lexibank/naganorgyalrongic.git@v2.0#egg=lexibank_naganorgyalrongic
27 -e git+https://github.com/lexibank/transnewguineaorg.git@v2.0#egg=lexibank_transnewguineaorg
28 -e git+https://github.com/lexibank/yanglalo.git@v2.0.1#egg=lexibank_yanglalo
29 -e git+https://github.com/lexibank/sohartmannchin.git@v2.0#egg=lexibank_sohartmannchin
30 -e git+https://github.com/lessersunda/lexirumah-data.git@v3.0.0#egg=pylexirumah
```

(3) Referenzkataloge, verteilte Daten, Metadaten

Daten stehen selten allein sondern sind auf externe Quellen bzw. Metadaten angewiesen.

Stabile (unter den vorangegangen Punkten kuratierte) Referenzkataloge erlauben mittels persistenter Referenzen (Identifier) das Einspeisen dieser Metadaten in sich in der Entwicklung befindliche Datensätze.



- Glottolog
 - Website, Datengrundlage
- Concepticon
 - Website, Datengrundlage
- CLTS
 - Website, Datengrundlage

Das gewählte Paketformat muss die Referenz auf diese Kataloge unterstützen und direkt einbetten.

```
"prov:wasDerivedFrom": [
    {
        "rdf:type": "prov:Entity",
        "dc:title": "Repository",
        "rdf:about": "https://github.com/lexibank/halenepal",
        "dc:created": "v2.0-23-g8ed928a"
    },
    {
        "rdf:type": "prov:Entity",
        "dc:title": "Glottolog",
        "rdf:about": "https://github.com/glottolog/glottolog/",
        "dc:created": "v4.3"
    },
    {
        "rdf:type": "prov:Entity",
        "dc:title": "Concepticon",
        "rdf:about": "https://github.com/concepticon/concepticon-data",
        "dc:created": "v2.4.0"
    },
    {
        "rdf:type": "prov:Entity",
        "dc:title": "CLTS",
        "rdf:about": "https://github.com/cldf-clts/clts/",
        "dc:created": "v2.0.0"
    }
]
```

```
{  
    "datatype": "string",  
    "propertyUrl": "http://cldf.clld.org/v1.0/terms.rdf#glottocode",  
    "valueUrl": "http://glottolog.org/resource/languoid/id/{glottolog_id}",  
    "name": "Glottocode"  
},
```

```
{  
    "datatype": "string",  
    "propertyUrl": "http://cldf.clld.org/v1.0/terms.rdf#concepticonReference",  
    "valueUrl": "http://concepticon.clld.org/parameters/{concepticon_id}",  
    "name": "Concepticon_ID"  
},
```

(4) Testsuiten für Daten

Ausgehend von der Grundannahme, dass Daten sich während ihrer Lebensphase stets wandeln können, bieten Tests (Modultests, Integrationstests, Validierungstests) ein sehr hilfreiches Werkzeug, um Fehler in Daten vorzubeugen bzw. Auswirkungen von Änderungen abschätzen zu können.

Dies kann in verschiedenen Granularitäten passieren, ist aber stets möglich und wertvoll.

Bewährt haben sich hier Dienste wie [Travis CI](#) oder aktuell [GitHub Actions](#).

Je nach gewünschtem Schärfegrad der Tests kann die Einbettung auch in komplexeren System wie [Buildbot](#) oder [Jenkins](#) stattfinden.

20 lines (12 sloc) | 616 Bytes

Raw Blame   

```
1 def test_valid(cldf_dataset, cldf_logger):
2     assert cldf_dataset.validate(log=cldf_logger)
3
4
5 def test_forms(cldf_dataset):
6     assert len(list(cldf_dataset["FormTable"])) == 3981
7     assert any(f["Form"] == "tánpe" for f in cldf_dataset["FormTable"])
8
9
10 def test_parameters(cldf_dataset):
11     assert len(list(cldf_dataset["ParameterTable"])) == 199
12
13
14 def test_languages(cldf_dataset):
15     assert len(list(cldf_dataset["LanguageTable"])) == 19
16
17
18 def test_cognates(cldf_dataset):
19     assert len(list(cldf_dataset["CognateTable"])) == 3769
20     assert any(f["Form"] == "porónno" for f in cldf_dataset["CognateTable"])
```

update new files

master lingulist -O- 535e3f2

✓ CLDF-validation
on: push

✓ build (3.6)

build (3.6)
succeeded on 25 Mar in 26s

Search logs ...

> ✓ Set up job 3s

> ✓ Run actions/checkout@v2 2s

> ✓ Set up Python 3.6 0s

> ✓ Install dependencies 6s

> ✓ Test with pytest 15s

1 ► Run pytest --cldf-metadata=cldf/cldf-metadata.json test.py
7 ===== test session starts =====
8 platform linux -- Python 3.6.13, pytest-6.2.2, py-1.10.0, pluggy-0.13.1
9 rootdir: /home/runner/work/halenepal/halenepal, configfile: setup.cfg, testpaths: test.py
10 plugins: cldf-0.2.1
11 collected 4 items
12
13 test.py [100%]
14
15 ===== 4 passed in 14.45s =====

> ✓ Post Run actions/checkout@v2 0s

Finished 6 days ago

Next →

Build steps	Build Properties	Worker: worker	Responsible Users	Changes	Debug
0 lexibank-uralex/20					4:43 build successful SUCCESS Triggered from: a-release-lexibank/14
0 worker_preparation					0 s worker ready
1 git					16 s update
2 virtualenv					4 s 'python3 -m ...'
3 upgrade tools					9 s './lexibank-uralex/bin/pip --cache-dir ...'
4 install pycldf					28 s './lexibank-uralex/bin/pip --cache-dir ...'
5 install dataset					43 s './lexibank-uralex/bin/pip --cache-dir ...'
6 install tools					6 s './lexibank-uralex/bin/pip --cache-dir ...'
7 makecldf					1:35 './lexibank-uralex/bin/cldfbench lexibank.makecldf ...'
8 pytest					30 s './lexibank-uralex/bin/pytest'
9 validate					24 s './lexibank-uralex/bin/cldf validate ...'
10 cldf check					3 s './lexibank-uralex/bin/cldf check ...'
11 cldfbench check					5 s './lexibank-uralex/bin/cldfbench --log-level ...'
12 lexibank check					10 s './lexibank-uralex/bin/cldfbench --log-level ...'
13 cldfbench diff					6 s './lexibank-uralex/bin/cldfbench diff ...'

(5) Nutzbarkeit von Daten während ihres gesamten Lebenszykluses

Datenerhebung und Datenanalyse ist ein iterativer Prozess.

All die vorgestellten Bausteine sind vor allem in einem Kontext sinnvoll, in dem Daten während ihres gesamten Lebenszykluses nutzbar sind (im Kontrast zu Daten die ein finales Artefakt darstellen sollen).

Ausgangspunkt dafür ist, dass das Erstellen bzw. Generieren **validen CLDFs** an erster Stelle steht.

Dadurch wird der iterative Prozess (einbinden und analysieren von CLDF in bestehende Pipelines, Verbesserung, Fehlersuche, neue Generierung, Versionierung, ...) erst ermöglicht.

Durch CLDF als menschen- und maschinenlesbares Format ist dies in jeder Phase möglich.

CLDF ist als tabellarisches Format einfach zu bearbeiten und zu sichten und kann gleichermassen aus einer Vielzahl von Formaten heraus erzeugt bzw. in eine Vielzahl von Formaten (SQL!) überführt werden.

MaerkangBolarGyalr...	MaerkangBolarGyalr...	1252_goodbye	lə22 las24	lə22_las24	l ə ² ² + l a s ² ⁴	Nagano2...
•MaerkangBolarGyalr...	MaerkangBolarGyalr...	1253_goodbye	tə22 na44 nŋo24	tə22_na44_nŋo24	t ə ² ² + n a ⁴ ⁴ + ...	Nagano2...
•MaerkangBolarGyalr...	MaerkangBolarGyalr...	1254_thankyou	nʒ22 ñar22 va44 / ...	nʒ22_ñar22_va44	n ʒ ² ² + ñ a r ² ² ...	Nagano2...
MaerkangBolarGyalr...	MaerkangBolarGyalr...	1255_comehere	ro22 ven44	ro22_ven44	r o ² ² + v e n ⁴ ⁴	Nagano2...
MaerkangBolarGyalr...	MaerkangBolarGyalr...	1256_look	kə22 nʒm22 cça22 ɾ...	kə22_nʒm22_cça22_ɾ...	k ə ² ² + n ʒ m ² ² ...	Nagano2...
1> forms						KEY_UP go-up 10085 rows •2

Links und Ressourcen

- CLDF:
 - <https://cldf.cldf.org/>
 - <https://github.com/cldf>
 - <https://github.com/cldf/cookbook>
 - <https://doi.org/10.1038/sdata.2018.205>
- Zenodo:
 - <https://zenodo.org/communities/cldf-datasets/>
 - <https://zenodo.org/communities/clics/>
 - <https://zenodo.org/communities/lexibank/>

- Reproduzierbare Analysen mit CLDF:
 - <https://doi.org/10.1038/s41597-019-0341-x>
 - <https://codeocean.com/capsule/7201165/tree>
- GitHub Organisationen:
 - <https://github.com/clld>
 - <https://github.com/concepticon>
 - <https://github.com/glottolog>
 - <https://github.com/lingpy>
 - <https://github.com/clics>